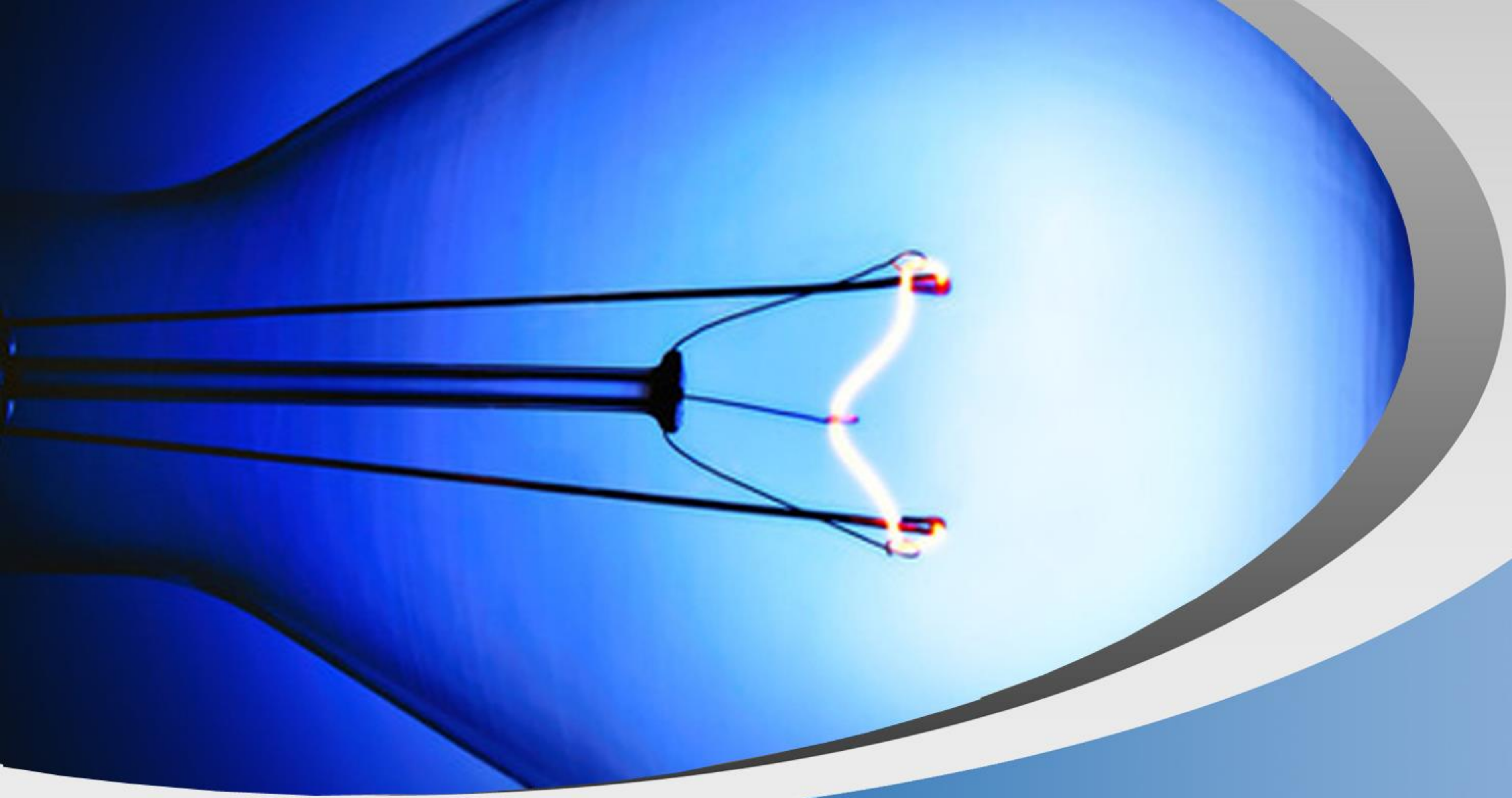


SystemC CCI WG
Indirectly Associating Parameters

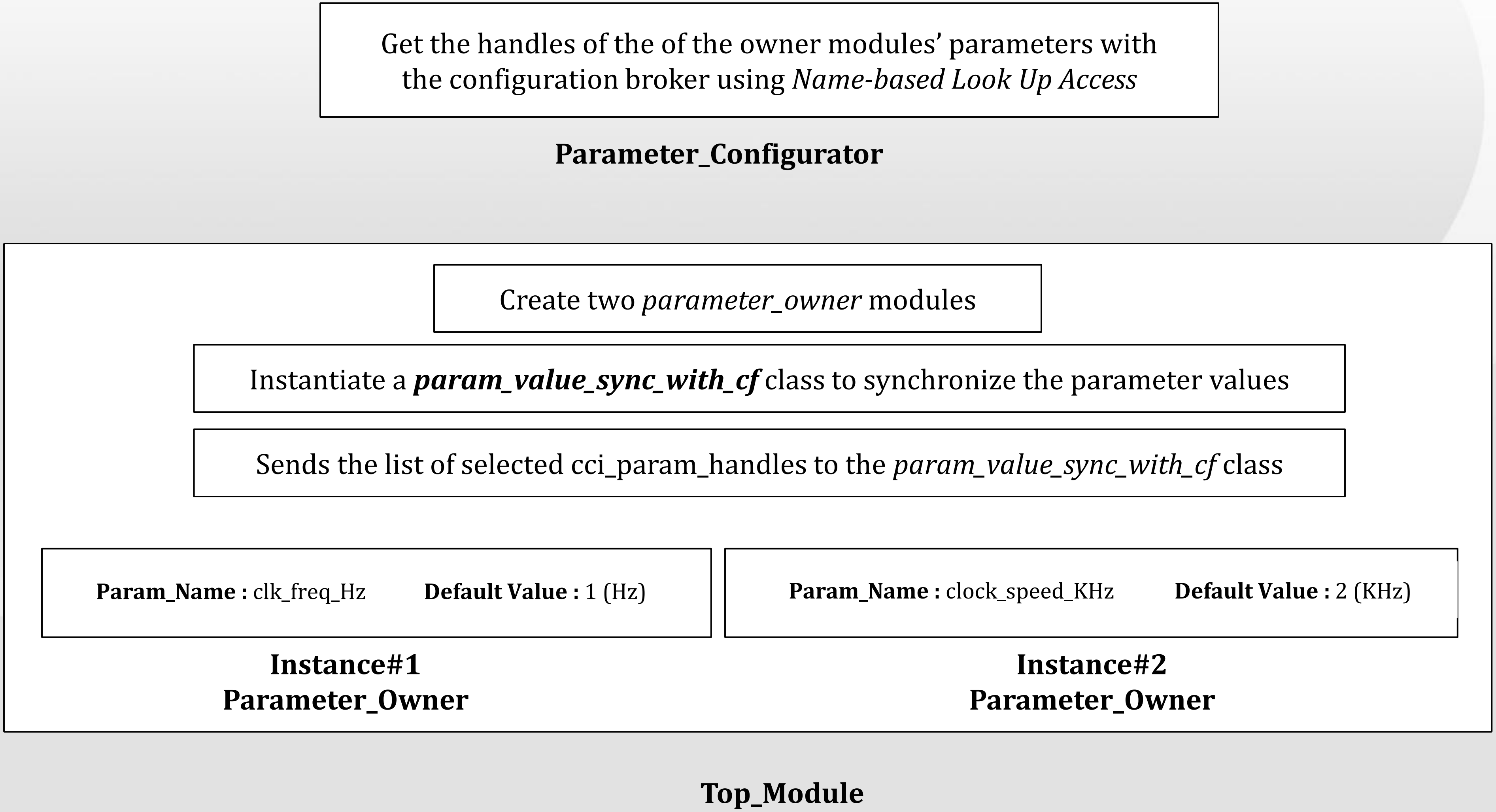
P V S Phaneendra, CircuitSutra Technologies Pvt. Ltd
August 2011

accellera
SYSTEMS INITIATIVE

Indirectly Associating Parameters

- **Objective of the present example is to demonstrate the following :**
  - Parameter Searches (R3), and
  - Parameter Value Synchronization to address Different Names, Same Intended Meaning (UC9)
  - value translation (e.g. units) is demonstrated between the associated parameters.

Example Illustration



Get the handles of the of the owner modules’ parameters with the configuration broker using *Name-based Look Up Access*

Parameter_Configurator

Create two *parameter_owner* modules

Instantiate a ***param_value_sync_with_cf*** class to synchronize the parameter values

Sends the list of selected cci_param_handles to the *param_value_sync_with_cf* class

Param_Name : clk_freq_Hz **Default Value :** 1 (Hz)

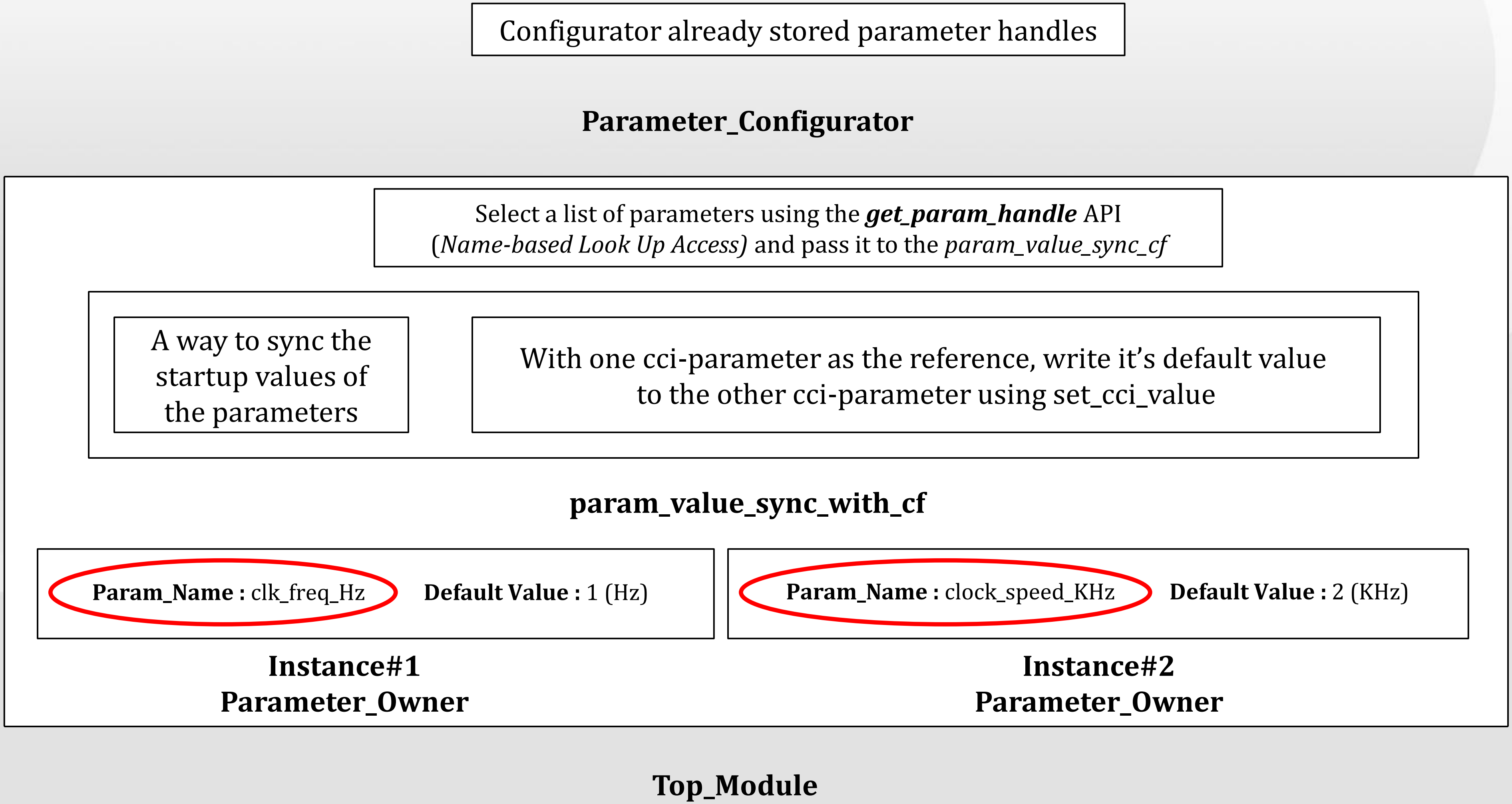
Instance#1
Parameter_Owner

Param_Name : clock_speed_KHz **Default Value :** 2 (KHz)

Instance#2
Parameter_Owner

Top_Module

Example Illustration Cont’d



Configurator already stored parameter handles

Parameter_Configurator

Select a list of parameters using the ***get_param_handle*** API (*Name-based Look Up Access)* and pass it to the *param_value_sync_cf*

A way to sync the startup values of the parameters

With one cci-parameter as the reference, write it’s default value to the other cci-parameter using set_cci_value

param_value_sync_with_cf

Param_Name : clk_freq_Hz **Default Value :** 1 (Hz)

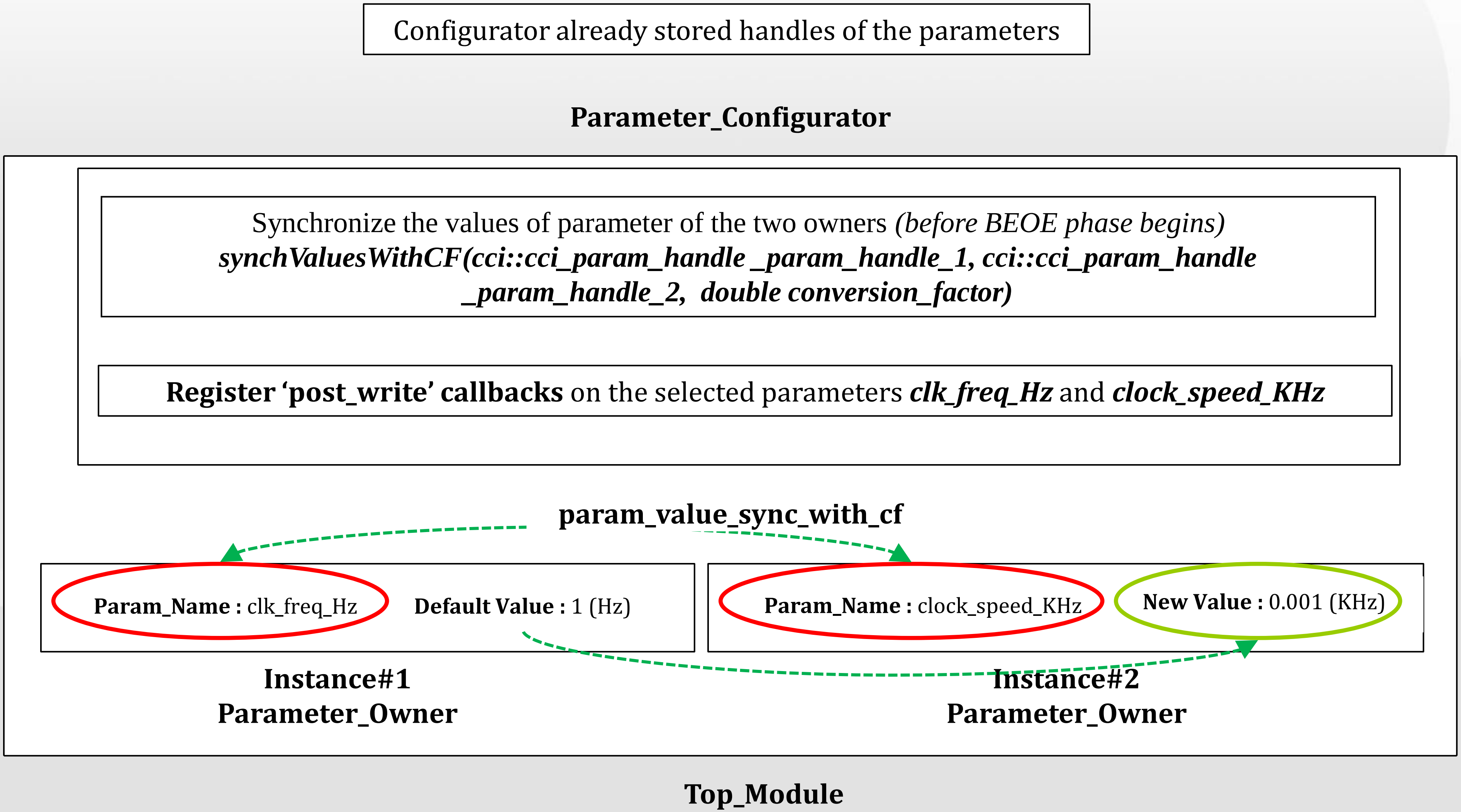
Instance#1
Parameter_Owner

Param_Name : clock_speed_KHz **Default Value :** 2 (KHz)

Instance#2
Parameter_Owner

Top_Module

Example Illustration Cont’d



Configurator already stored handles of the parameters

Parameter_Configurator

Synchronize the values of parameter of the two owners *(before BEOE phase begins)*
synchValuesWithCF(cci::cci_param_handle _param_handle_1, cci::cci_param_handle _param_handle_2, double conversion_factor)

Register ‘post_write’ callbacks on the selected parameters ***clk_freq_Hz*** and ***clock_speed_KHz***

param_value_sync_with_cf

Param_Name : clk_freq_Hz **Default Value :** 1 (Hz)

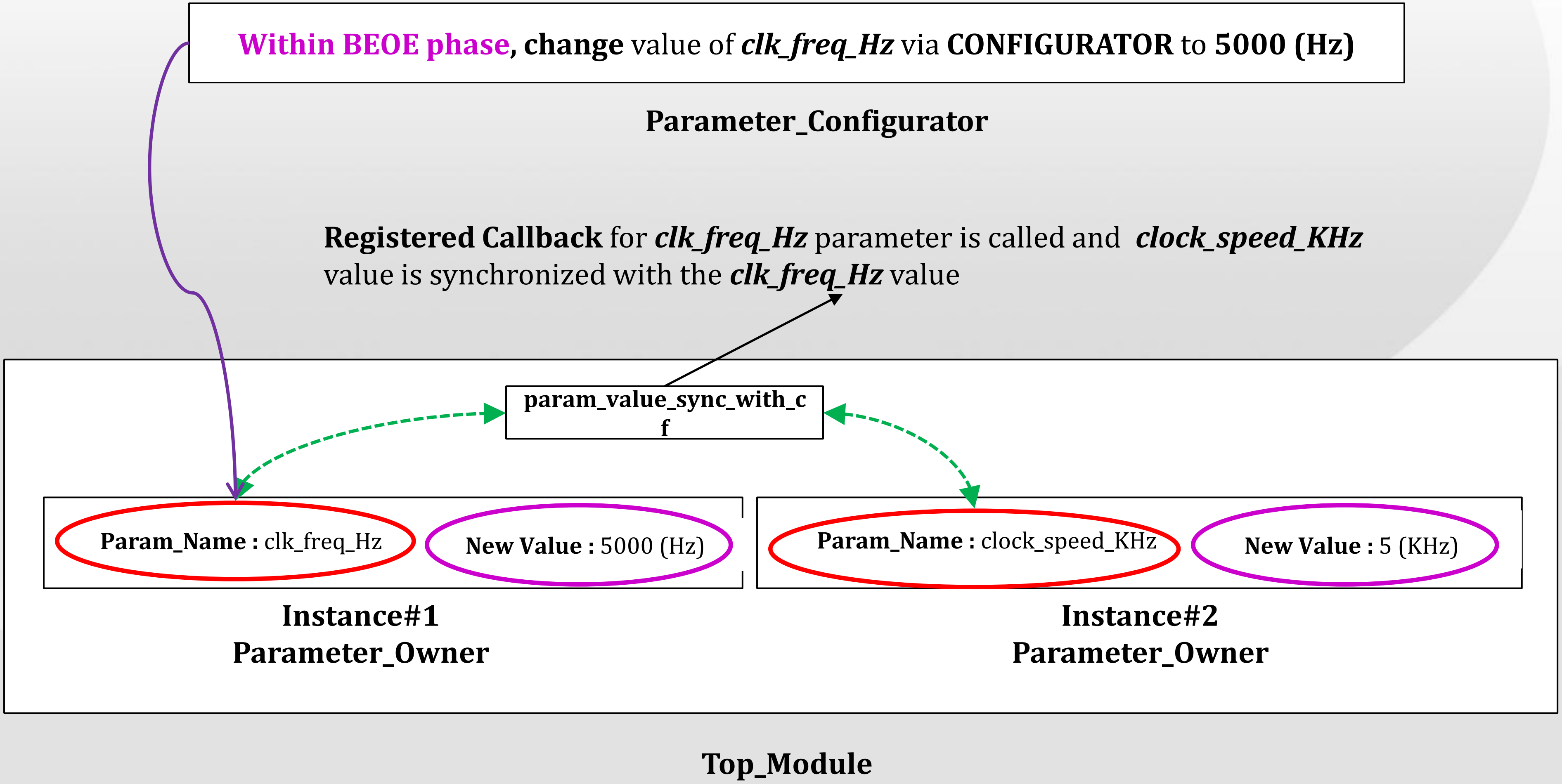
Instance#1
Parameter_Owner

Param_Name : clock_speed_KHz **New Value :** 0.001 (KHz)

Instance#2
Parameter_Owner

Top_Module

Example Illustration Cont’d



Within BEOE phase, **change** value of ***clk_freq_Hz*** via **CONFIGURATOR** to **5000 (Hz)**

Parameter_Configurator

Registered Callback for ***clk_freq_Hz*** parameter is called and ***clock_speed_KHz*** value is synchronized with the ***clk_freq_Hz*** value

param_value_sync_with_cf

Param_Name : clk_freq_Hz

New Value : 5000 (Hz)

Param_Name : clock_speed_KHz

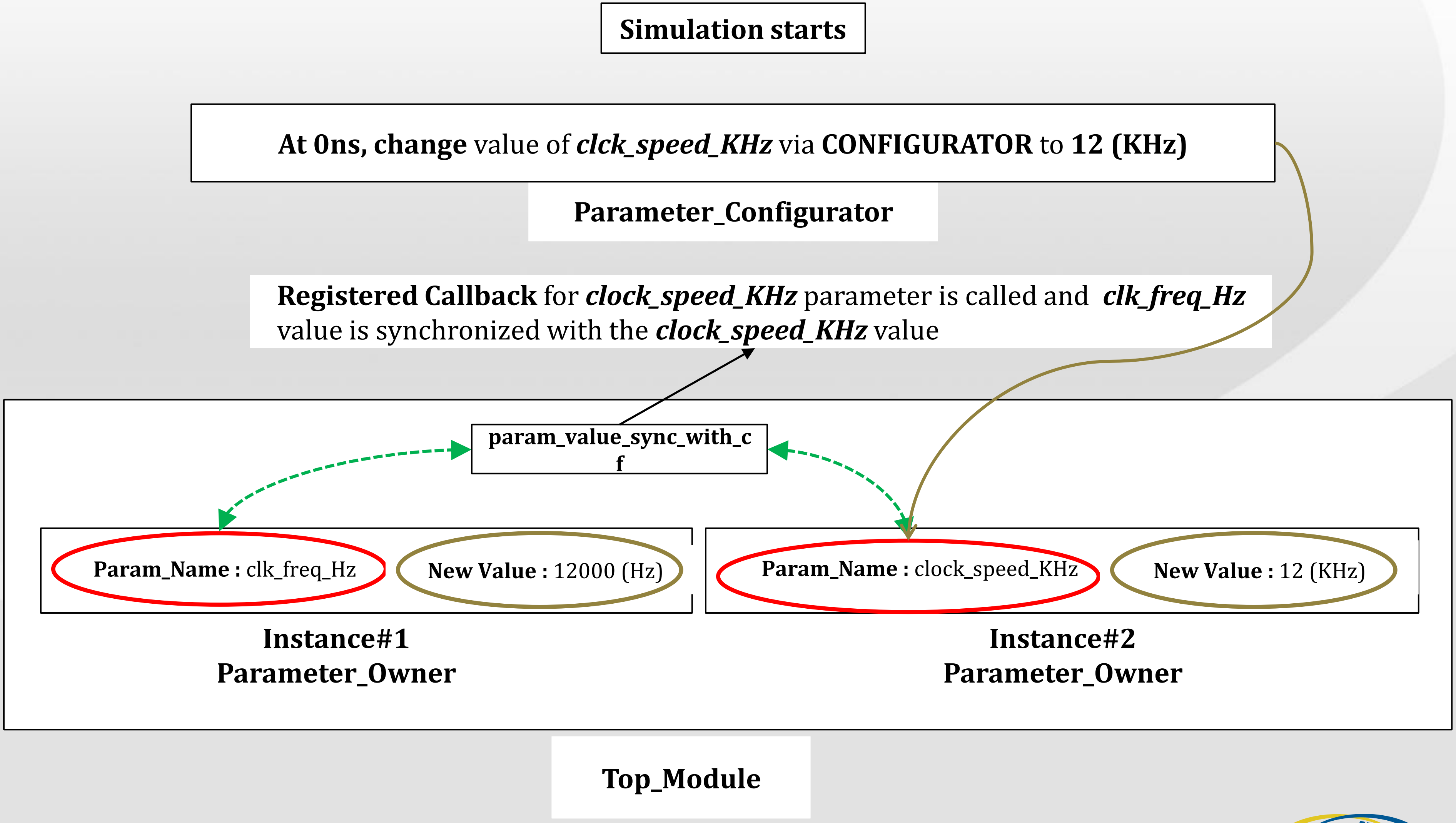
New Value : 5 (KHz)

Instance#1
Parameter_Owner

Instance#2
Parameter_Owner

Top_Module

Example Illustration Cont’d



Simulation starts

At 0ns, change value of ***clck_speed_KHz*** via **CONFIGURATOR** to **12 (KHz)**

Parameter_Configurator

Registered Callback for ***clock_speed_KHz*** parameter is called and ***clk_freq_Hz*** value is synchronized with the ***clock_speed_KHz*** value

param_value_sync_with_cf

Param_Name : clk_freq_Hz

New Value : 12000 (Hz)

Param_Name : clock_speed_KHz

New Value : 12 (KHz)

Instance#1
Parameter_Owner

Instance#2
Parameter_Owner

Top_Module

Expected Output
(ex12_Indirectly_Associateing_Parameters.log)

```
SystemC Simulation

Info: top_mod.param_owner1: @0 s, [OWNER C_TOR] : Parameter Name    : top_mod.param_owner1.clk_freq_Hz,
Value : 1

Info: top_mod.param_owner2: @0 s, [OWNER C_TOR] : Parameter Name    :
top_mod.param_owner2.clock_speed_KHz, Value : 2

Info: top_mod: @0 s, [TOP_MODULE C_TOR] : Parameter Name : top_mod.param_owner1.clk_freq_Hz, Value : 1.0

Info: top_mod: @0 s, [TOP_MODULE C_TOR] : Parameter Name : top_mod.param_owner2.clock_speed_KHz, Value :
2.0

Info: top_mod.param_value_sync_with_cf: @0 s, Parameter1_str: top_mod.param_owner1.clk_freq_Hz

Info: top_mod.param_value_sync_with_cf: @0 s, Parameter2_str : top_mod.param_owner2.clock_speed_KHz

Info: top_mod.param_value_sync_with_cf: @0 s, ConversionFactor : 0.001

Info: param_cfgr: @0 s, [CFGR C_TOR] : Parameter Name : top_mod.param_owner1.clk_freq_Hz, Value : 1.0

Info: param_cfgr: @0 s, [CFGR C_TOR] : Parameter Name : top_mod.param_owner2.clock_speed_KHz, Value :
0.001

Info: sc_main: Begin Simulation.

Info: param_cfgr: @0 s, [CFGR within beoe] Within the BEOE phase

Info: param_cfgr: @0 s, [CFGR within beoe] : Changing the 'clk_freq_Hz' of OWNER (1) to 5000 (Hz).
```

Cont’d

```
Info: top_mod.param_value_sync_with_cf: @0 s, [PARAM_VALUE_SYNC - post_write callback] : Parameter Name
: top_mod.param_owner1.clk_freq_Hz, Value : 5000.0

Info: top_mod.param_value_sync_with_cf: @0 s, [PARAM_VALUE_SYNC - post_write callback] : Parameter Name
: top_mod.param_owner2.clock_speed_KHz, Value : 5.0

Info: param_cfgr: @0 s, [CFGR within beoe] : Parameter Name : top_mod.param_owner1.clk_freq_Hz, Value :
5000.0

Info: param_cfgr: @0 s, [CFGR within beoe] : Parameter Name : top_mod.param_owner2.clock_speed_KHz,
Value : 5.0

Info: param_cfgr: @0 s, @ 0 s

Info: param_cfgr: @0 s, [CFGR] : Changing the 'clock_speed_KHz' of OWNER (2) to 12 (KHz).

Info: top_mod.param_value_sync_with_cf: @0 s, [PARAM_VALUE_SYNC - post_write callback] : Parameter Name
: top_mod.param_owner2.clock_speed_KHz, Value : 12.0

Info: top_mod.param_value_sync_with_cf: @0 s, [PARAM_VALUE_SYNC - post_write callback] : Parameter Name
: top_mod.param_owner1.clk_freq_Hz, Value : 12000.0

Info: param_cfgr: @0 s, [CFGR] : Parameter Name : top_mod.param_owner1.clk_freq_Hz, Value : 12000.0

Info: param_cfgr: @0 s, [CFGR] : Parameter Name : top_mod.param_owner2.clock_speed_KHz, Value : 12.0

Info: sc_main: End Simulation.
```